Eye contact detection

Bachelor Semester Project

Lucas Strauss

lucas.strauss@epfl.ch

Visual Intelligence for Transportation (VITA) lab
EPFL

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Supervisors:
Liu Yuejing
Prof. Alexandre Alahi

January 11, 2019

1 Introduction

Eye contact is a regular non-verbal social behavior that we use in our everyday life. One of the main use is to gather feedback [1] to see how the information was received by the counterpart. An eye contact detector can have many uses in human-computer interaction, but commercial equipment are expensive, require dedicated hardware and sometimes personal calibration. Here we study a real-time machine learning algorithm that uses a webcam stream to detect the eye contact with the camera.

This project converts an initial gaze estimation algorithm into an eye contact detector. First we evaluate the algorithm using a gaze dataset to address its weaknesses, then we propose another method to try to improve the results.

2 Related work

This work is mainly based on the study of Seonwook Park [4] and its publicly available algorithm[1] from the ETHZ. This project is explained in details in the next section.

Smith et al. [5] use a similar approach but instead of estimating the gaze direction they trained their network to be a binary classifier to indicate whether the person is making eye contact with the camera or not. Unfortunately their algorithm is not publicly available. However, they created a gaze dataset that is used in this project to evaluate the algorithm.

Unlike previous research, such as Morimoto et al.'s [3], these methods work without any calibration and at medium and close range. This is particularly useful when working in a crowded space because we do not have any information about the targets.

Another recent work from Xucong Zhang et al. [6] from the Max-Plank Institute for Informatics is worth mentioning. He and his team developed a CNN that takes multiple input parameters in addition to the face image such as the head pose orientation. They normalize the input images to handle different hardware setups, which our current algorithm does not. Unfortunately, the code of this project is not publicly available either.

3 Method

The pipeline works as showed in Fig.1
The face detection step is using the *dlib*[2] library which uses an improved HOG[3] combined with a linear classifier. This method seems to work fine [2] with clean images but as showed in section 4, some problems still remain.

[1] `https://ait.ethz.ch/projects/2018/landmarks-gaze`
[2] `http://dlib.net`
[3] Histogram of Oriented Gradients

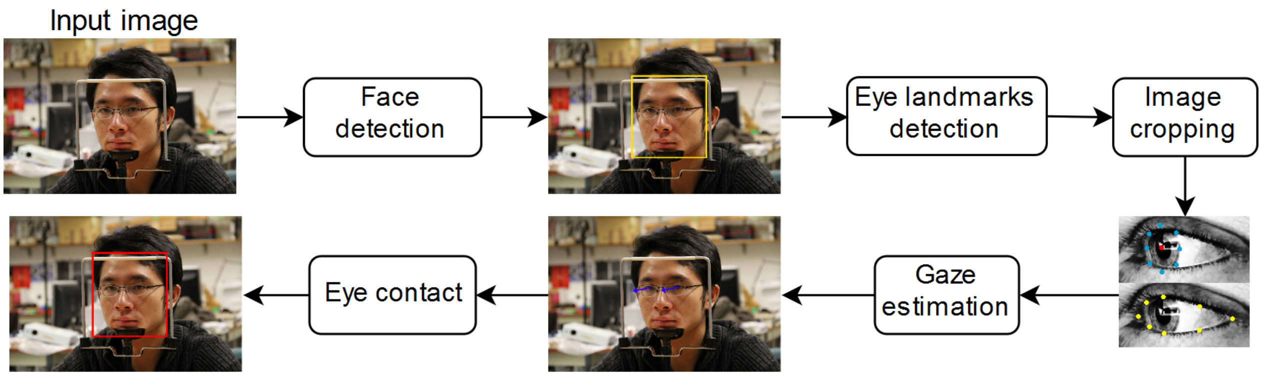

Figure 1: Pipeline of the algorithm

The facial landmarks are detected using the *dlib* library too. An HOG with linear classifier was trained on the iBug facial points dataset[4]. Only the eye landmarks are taken into account here. The image is cropped using the leftmost and rightmost landmark for each eye.

The heart of this algorithm is the gaze estimator. It features a SVR (Support Vector Regression) machine that takes 18 2D points: 8 points from the iris edge, 8 points from the eyelids edge, the iris center and a raw estimation of the gaze obtained by subtracting the eyeball center by the iris center. The eyeball center is computed as the midpoint between the leftmost eye landmark and the rightmost one. The output is a simple vector (ϕ, θ) representing the horizontal and vertical angle respectively.
Once the gaze direction is obtained a simple threshold is applied to determine if the person is looking at the camera or not.

$$\phi < threshold \textbf{ and } \theta < threshold$$

A low threshold gives a higher accuracy but less realistic results in real-time performance (see section 4).

4 Evaluation

To evaluate the algorithm we generated some videos of random images from the Columbia Gaze dataset [5] that was used as input for the algorithm. More details about this step can be found in the wiki of the Github repository of this project[5].

First, we manually evaluate the face detection algorithm by running only the face detection part of the algorithm against 500 hundred frames. This revealed an accuracy of 78%.

[4]see `https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/`
[5]`https://github.com/VFXOne/eye-contact`

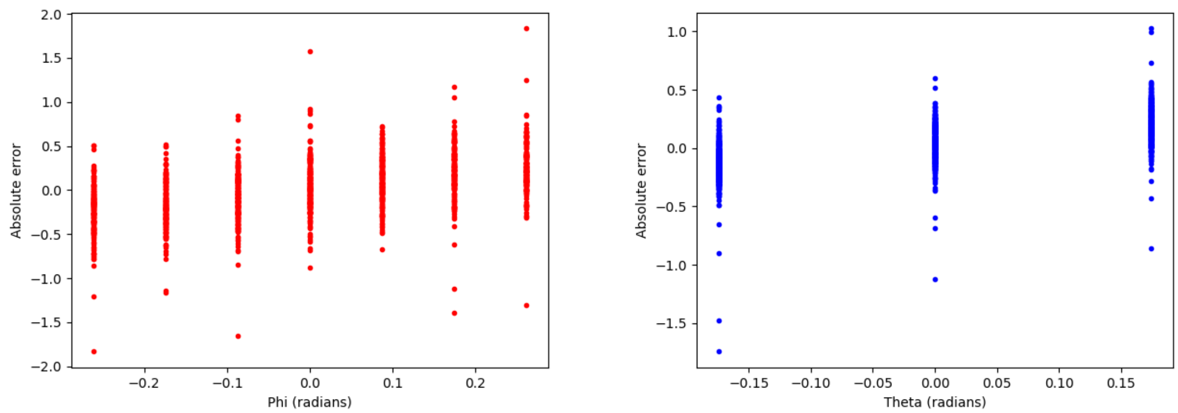

Figure 2: Absolute error of ϕ and θ angles

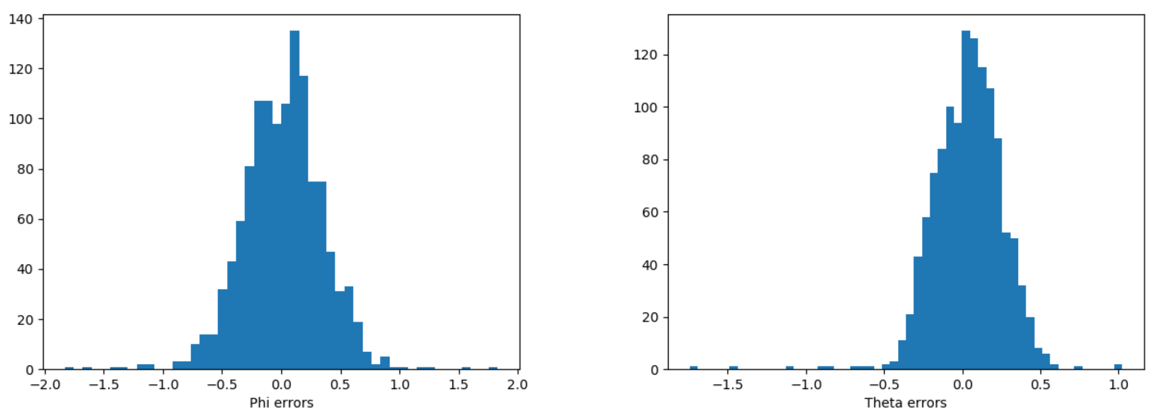

Figure 3: Histogram of ϕ and θ errors

With the exception of a few outliers caused by the face detector's lack of accuracy, the algorithm's precision rate is ± 30 degrees (± 0.5 radian) for the vertical gaze angle. Note that the Columbia Gaze dataset features only 3 different vertical angles and 7 different horizontal angles. The vertical angle estimation is slightly better than the horizontal gaze angle estimation whose absolute error lies between -60 and $+60$ degrees (± 1 radian). This is worse than the precision of 16 degrees mentioned by Seonwook Park [4] in his paper.

An attempt to evaluate the algorithm with the MPIIGaze [7] dataset has been made but unfortunately, the dataset only features cropped face images like Fig.4 and the current algorithm requires the full face to be visible to detect the face and eye landmarks.

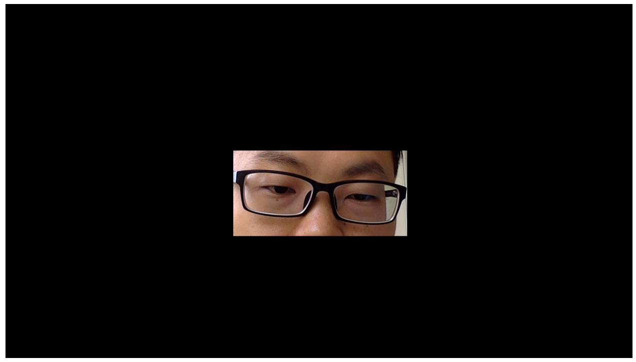

Figure 4: An example image of the MPIIGaze dataset

We tried to replace the *dlib* face detector by the DNN face detector of *OpenCV* [6] but the results only show a slight improvement (as you can see in Fig.7 and Fig.8), giving almost the same precision for both angles.

Even if the face detection step can be improved and the gaze estimator is not very precise, our evaluations showed that the eye contact detection can still achieve very good results as showed in Fig.9. Clearly the accuracy decreases as the threshold increases. But a low threshold seems to give less realistic results in real-time performance. Indeed, one must be closer to the camera and stare longer at it in order to enable an eye contact detection by the algorithm. A threshold of 3 degrees seems to be a good compromise between accuracy and realism.

5 Future work

In this project we evaluated the performance of a gaze estimation algorithm for eye contact detection. The lack of precision is mostly due to the face detector and the gaze estimator. Mask R-CNN and OpenPose have been suggested to replace the current face detector. While these algorithms are state of the art in human pose detection, they are trained to estimate the full body pose. Because we only care about the face landmarks this might result in high latency and would not be suitable for real-time performance.

The current method to detect the eye contact relies on a regression algorithm while the eye contact is a classification problem. Directly training an algorithm to classify the eyes as "eye contacting" or not, as done in [5], would probably lead to better results.

[6] `https://github.com/opencv/opencv/tree/master/samples/dnn`

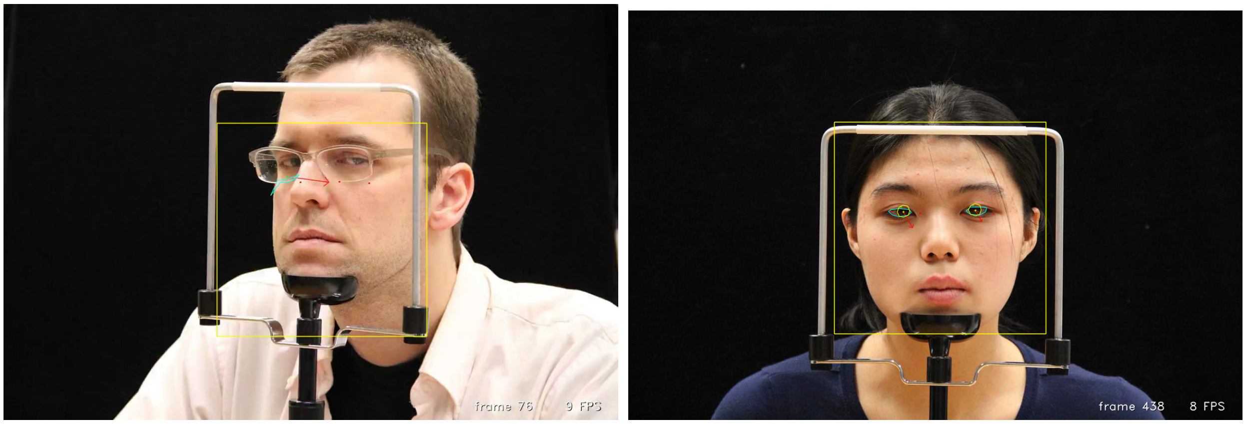

Figure 5: One of the worst and one of the best output of the algorithm

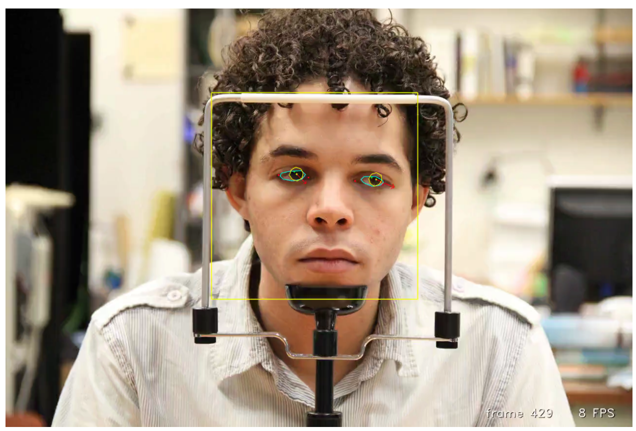

Figure 6: An average image output

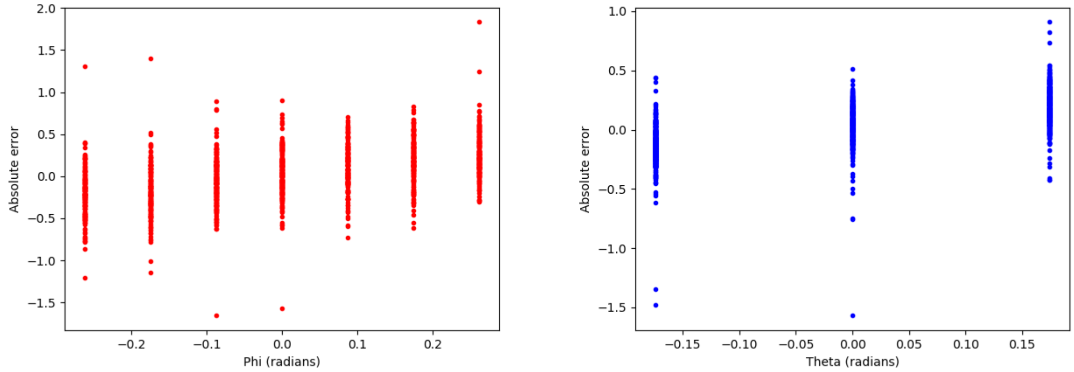

Figure 7: Absolute error of ϕ and θ angles using OpenCV's face detector

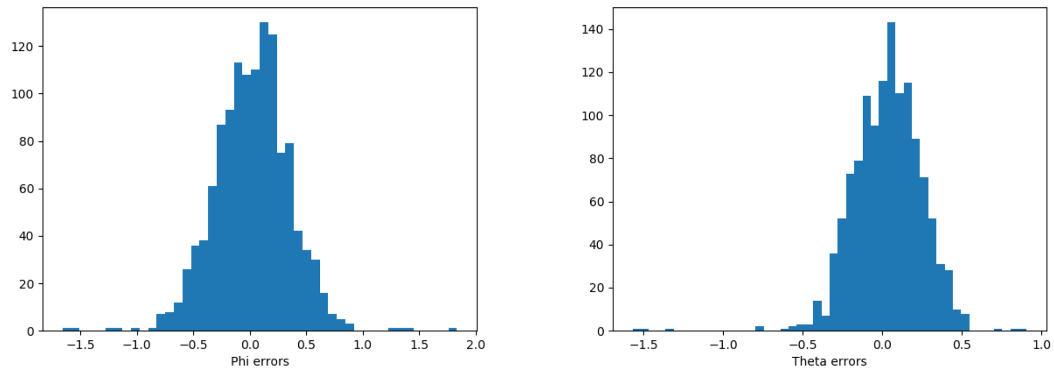

Figure 8: Histogram of ϕ and θ errors using OpenCV's face detector

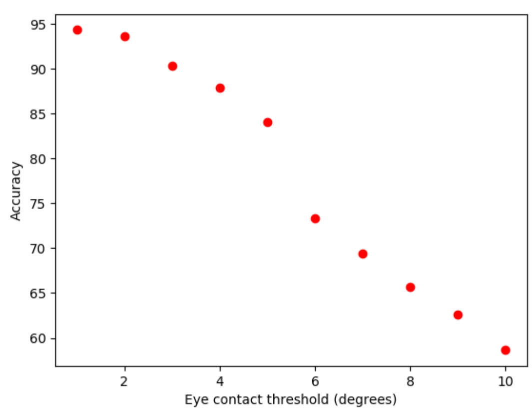

Figure 9: The accuracy of the eye contact detection as the threshold increases

References

[1] Michael Argyle and Janet Dean. “Eye-contact, distance and affiliation”. In: *Sociometry* (1965), pp. 289–304.

[2] Vikas Gupta. *Face Detection – OpenCV, Dlib and Deep Learning (C++ / Python)*. 2018. URL: `https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python` (visited on 01/06/2018).

[3] Carlos H Morimoto, Arnon Amir, and Myron Flickner. “Detecting eye position and gaze from a single camera and 2 light sources”. In: IEEE. 2002, p. 40314.

[4] Seonwook Park et al. “Learning to find eye region landmarks for remote gaze estimation in unconstrained settings”. In: *arXiv preprint arXiv:1805.04771* (2018).

[5] Brian A Smith et al. “Gaze locking: passive eye contact detection for human-object interaction”. In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM. 2013, pp. 271–280.

[6] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. “Everyday eye contact detection using unsupervised gaze target discovery”. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM. 2017, pp. 193–203.

[7] Xucong Zhang et al. “Appearance-based gaze estimation in the wild”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4511–4520.